

Patent Application

for

**GEOGRAPHICAL DATA MARKUP ON A PERSONAL DIGITAL
ASSISTANT (PDA)**

Inventor(s):

EDWARD J. CONNOR
NEMMARA CHITHAMBARAM
JOHN RICARDO DEAGUIAR

Prepared By:

Jason S. Feldmar
Gates & Cooper
Howard Hughes Center
Suite 1050
6701 Center Drive West
Los Angeles, California 90045

09628851.073100

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit under 35 U.S.C. §119(e) of the following co-pending and commonly assigned U.S. Provisional patent applications, which applications are incorporated by reference herein:

5 United States application serial number 60/159,069, entitled "MAPGUIDE PERSONAL DIGITAL ASSISTANT," filed on October 12, 1999, by Nemmara Chithambaram, et. al., attorney's docket number 30566.96USP1;

 United States application serial number 60/193,141, entitled "SHARABLE SERVER UPLOADABLE REDLINING FOR PERSONAL DIGITAL
10 ASSISTANT (PDA)," filed on March 29, 2000, by Nemmara Chithambaram, et. al., attorney's docket number 30566.97USP1;

 United States application serial number 60/193,153 entitled "INDEXED RASTER VECTOR SCHEMA FOR PERSONAL DIGITAL ASSISTANT
15 (PDA) DATABASES," filed on March 29, 2000, by Nemmara Chithambaram, et. al., attorney's docket number 30566.109USP1;

 United States application serial number 60/193,142, entitled
"INTERPROCESS API AND GRAPHICAL USER INTERFACE FOR
PERSONAL DIGITAL ASSISTANT (PDA) DEVICES AND
APPLICATIONS," filed on March 29, 2000, by Nemmara Chithambaram, et. al.,
20 attorney's docket number 30566.110USP1; and

 United States application serial number 60/193,862, entitled "MAPGUIDE FOR MOBILE DEVICES SERVER," filed on March 30, 2000, by Nemmara Chithambaram, et. al., attorney's docket number 30566.112USP1.

This application is related to the following co-pending and commonly-assigned patent applications, which applications are incorporated by reference herein:

5 United States Patent Application Serial No. 09/411,506, entitled
“VECTOR-BASED GEOGRAPHIC DATA”, by Gregory A. Roy, et. al.,
Attorney Docket No. 30566.17USC1, filed on October 4, 1999, which is a
continuation patent application of United States Patent No. 5,966,135 issued on
October 12, 1999 (Application Serial No. 08/757,706 filed on October 30, 1996),
10 by Gregory A. Roy et al., entitled “VECTOR-BASED GEOGRAPHIC DATA”;

United States Patent Application Serial No. xx/xxx,xxx, entitled
“METHOD AND APPARATUS FOR PROVIDING ACCESS TO MAPS ON
A PERSONAL DIGITAL ASSISTANT (PDA)”, by Nemmara Chithambaram et
al., Attorney Docket No. 30566.96USU1, filed on the same date herewith;

15 United States Patent Application Serial No. xx/xxx,xxx, entitled
“GENERALIZED, DIFFERENTIALLY ENCODED, INDEXED RASTER
VECTOR DATA AND SCHEMA FOR MAPS ON A PERSONAL DIGITAL
ASSISTANT”, by Nemmara Chithambaram et al., Attorney Docket No.
30566.98USU1, filed on the same date herewith; and

20 United States Patent Application Serial No. xx/xxx,xxx, entitled
“METHOD AND APPARATUS FOR OBTAINING A SET OF MAPS”, by
Nemmara Chithambaram et al., Attorney Docket No. 30566.112USU1, filed on
the same date herewith.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

5 The present invention relates generally to electronic maps and geographic information, and in particular, to a method, apparatus, and article of manufacture for marking up and redlining maps and geographic data on a personal digital assistant (PDA).

2. Description of the Related Art

10 Computer implemented geographic information systems (GIS) are known in the art. Such GIS provide for the retrieval and display of geographic information (e.g., maps). Additionally, the ability to redline or mark up an architectural drawing is known in the art. Such redlining capabilities allow a user to superimpose or draw a shape, text, or other information on an existing drawing.

15 However, the use of a GIS on a personal digital assistant (PDA) and marking up or redlining a map displayed on a PDA are not known or disclosed in the art.

Field/utility technicians such as gas company employees, salespersons, plumbers, insurance adjusters, or any type of employment that requires travel to different locations, often utilize or require access to maps and geographic
20 information. Further, such technicians often need to interact with and markup or redline a map to refer to at a later time. For example, a plumber/contractor may want to determine where the main gas line or water line on a street is located. Additionally, if the main gas line or water line is not in the location specified in the

001E70-T5882960

map (or such data is not available), the plumber/contractor may want to markup the map to indicate where the gas line or water line is located. While out in the field, the technicians often do not have a network connection, and carrying a laptop or desktop computer is cumbersome and impractical. Thus, it is desirable to
5 have a small (handheld) portable computing device with the capabilities to display, interact, and markup geographic information both online and offline.

Prior art handheld computing devices (also referred to as palm PCs or personal digital assistants (PDAs)), are often used to access and utilize personal information. Many handheld computing devices are available in today's
10 marketplace. Typically, handheld computing devices are only slightly larger than the size of one's palm (hence, the name palm PC) and have a small display screen for viewing a plethora of items. Software can be installed on a PDA to provide enhanced functionality. For example, a personal productivity tool may be installed to provide access to a calendar, contacts, email, Internet browsing, audio books,
15 and audio recording capabilities. Card slots may also be available to provide additional memory or other functions (e.g., a modem). Additionally, some PDAs have infrared (IR) ports for communication.

The PDA environment, however, poses several challenges for geographic information systems and markup/redlining in terms of memory, storage, processor
20 speeds, wireless transmission rates, and display attributes. For example, PDAs commonly only maintain 96K or less of memory, 2 Mb (megabytes) or less storage, a 13 MHz processor speed, and a black and white or gray scale display mechanism. Displaying a map and maintaining the ability to redline or markup a map on a

PDA must be within such limitations.

Thus, there is a need for a geographic information system that overcomes the above described deficiencies on a portable handheld electronic device such as a PDA.

5

SUMMARY OF THE INVENTION

One or more embodiments of the invention provide for redlining or marking up geographic information on a personal digital assistant (PDA).

Embodiments include the ability to synchronize markup data on a PDA with a server and the graphical user interface for creating markup data. The PDA maintains the functionality commonly available in a standard client comprised of a complete computer system. The PDA provides uploadable, sharable redlining data (created from scribbles on the field).

PDA clients can markup the map data that is downloaded with each mapset. The markup graphics elements include a redline and a text note. The redline is a zero width vector that mimics ink flowing from a stylus, and an associated text note that pops up as a tooltip when the object is selected. The text note provides text blocks, represented on the map as note symbols anchored at a point, and an associated text note that pops up as a tooltip when the object is selected. Any number of these objects can be included in the markup layer of a specific mapset. Each mapset references a single markup layer.

Markup layers are stored locally and uploaded (synchronized) with the server automatically when the mapset list is updated. The markup layer, being a

component of the mapset's data cache, either remains on the client or is purged with the mapset. If the mapset and associated markup layer stay on the client, subsequent edits may be made to it. These edits are copied up to the server at the next synchronization. Once a mapset leaves the PDA device, markup objects leave
5 with it. When the map is accessed again, the note or redline object may or may not be with it, depending on how this map is handled on the server.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent
10 corresponding parts throughout:

FIG. 1 schematically illustrates a hardware and software environment for the architecture in accordance with one or more embodiments of the invention;

FIG. 2 illustrates a high level architecture that enables redlining and markup processing capabilities in accordance with one or more embodiments of the
15 invention;

FIG. 3 is a flow chart illustrating the basic data flow for processing personal digital assistant markup data in accordance with one or more embodiments of the invention;

FIG. 4 is a flow chart illustrating a synchronization application markup
20 upload process in accordance with one or more embodiments of the invention;

FIG. 5 is a flow chart illustrating the creation of a redline in accordance with one or more embodiments of the invention;

FIG. 6A illustrates a redline object on a personal digital assistant device in

accordance with one or more embodiments of the invention;

FIG 6B illustrates a text entry box for a redline object in accordance with one or more embodiments of the invention; and

FIG. 7 is a flow chart illustrating the creation of a markup note object in
5 accordance with one or more embodiments of the invention.

09628851.073100

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description, reference is made to the accompanying drawings which form a part hereof, and which is shown, by way of illustration, several embodiments of the present invention. It is understood that other
5 embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

Overview

One or more embodiments of the invention provide for redlining and
10 markup capabilities on a geographical information system (GIS) implemented on a personal digital assistant (PDA). Software on the PDA is enabled to provide such functionality. PDA users (e.g., technicians) can make scribbles and annotations on a map displayed on a PDA using a paper and pencil metaphor. To accommodate such redlining, embodiments provide a redline object that consists of geometric
15 scribbles (points, lines, polygons, symbols), GPS (global positioning system) input coordinates, annotations (positioned text with font information, etc.), and a geo-reference system that allows the redline object to be integrated into a spatial database using a server and displayed on other maps. The redlining system captures user input and allows for redline objects to be uploaded to a server. Further, the
20 level of sharing on the server side is configurable (e.g., personal, group, global, etc.).

Accordingly, PDA clients can markup the map data that is downloaded with each mapset. The markup graphics elements include a georeferenced polyline redline and a georeferenced text note. The georeferenced polyline redline is a zero

width vector that mimics ink flowing from a stylus, and an associated text note that pops up as a tooltip when the object is selected. The georeferenced text note provides text blocks, represented on the map as note symbols anchored at a point, and an associated text note that pops up as a tooltip when the object is selected.

- 5 Any number of these objects can be included in the markup layer of a specific mapset. Each mapset references a single markup layer.

Markup layers are stored locally and uploaded (synchronized) with the server automatically when the mapset list is updated. The markup layer, being a component of the mapset's data cache, either remains on the client or is purged
10 with the mapset. If the mapset and associated markup layer stay on the client, subsequent edits may be made to it. These edits are copied up to the server at the next synchronization. Once a mapset leaves the PDA device, markup objects leave with it. When the map is accessed again, the note or redline object may or may not be with it, depending on how this map is handled on the server.

15

General Architecture

Hardware Environment

FIG. 1 schematically illustrates a hardware and software environment for the architecture in accordance with one or more embodiments of the invention. A
20 typical distributed computer system 100 uses a network/Internet 118 to connect technicians utilizing clients such as a thin client 102 (e.g. a PDA, WINCE, or PALM device) or a thick client 104 (e.g., a computer system running a browser) to server computers 106.

personal computers, workstations, minicomputers, or mainframes.

09628851.073100

The network/Internet 118 connects client computers 102 and 104 executing the appropriate software applications to server computers 106 executing Web servers 110, MapGuide servers 120, and servlets 108. MapGuide servers 120 and servlets 108 may be located within or part of Web server 110. The server 106 and its components may also be referred to as a back office system. Such a back office system maintains access to corporate databases, synchronization utilities, etc. The Web server 110 is typically a program such as IBM's HyperText Transport Protocol (HTTP) Server or Microsoft's Internet Information Server. The servlet 108 communicates with thin client 102 through Web server 110 such that any additional processing required by a thin client 102 may be performed by the servlet 108. The servers 106 also execute a Common Gateway Interface (CGI) 112 (or Netscape Application Programming Interface (NSAPI), Internet Server Application Programming Interface (ISAPI), etc.), which interfaces between the Web server 110 and a database management system (DBMS) 114 that may be utilized to retrieve relevant geographical data (such as SDF data, raster data, Open DataBase Connectivity (ODBC) data, etc.) from database 116.

Generally, components 108-116 and 120-128 all comprise logic and/or data that is embodied in or retrievable from a device, medium, signal, or carrier, e.g., a data storage device, a data communications device, a remote computer or device coupled to the computer via a network or via another data communications device, etc. Moreover, this logic and/or data, when read, executed, and/or interpreted, results in the steps necessary to implement and/or use the present invention being performed.

Thus, embodiments of the invention may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" (or alternatively, 5 "computer program product") as used herein is intended to encompass logic and/or data accessible from any computer-readable device, carrier, or media.

Those skilled in the art will recognize many modifications may be made to this exemplary environment without departing from the scope of the present invention. For example, those skilled in the art will recognize that any 10 combination of the above components, or any number of different components, including different logic, data, different peripherals, and different devices, may be used to implement the present invention, so long as similar functions are performed thereby. Specifically, those skilled in the art will recognize that the present invention may be applied to any database, associated database management 15 system, or peripheral device.

Software Architecture

In accordance with the hardware descriptions, thick clients 104 are complete computer systems with web browsers and full processing capabilities. However, 20 the hardware limitations of a PDA device necessitate software limitations. Accordingly, to enable a geographic information system and to enable redlining and markup capabilities on a PDA device, a thin client 102 is provided. To accommodate the thin client, additional support on server 106 may be utilized.

For example, server 106 may provide for the conversion of data, more generalized display data, simplified project files, de-cluttering services, and possibly server management of the user state.

5 The architecture of the invention includes a data model that combines static raster layers (static raster data for multiple layers) with live vector objects to deliver good display and download performance, and also provides interactive selectable objects.

Vector based maps (also referred to as “map layer data” consisting of geographic information/data for one or more layers) are served by a servlet 108 and
10 are an encoded and spatially indexed vector representation of the geographic data. Such vector maps provide for a more “interactive” display with flexible zooming on the client 102, highlighting, etc. Alternatively, the Scalable Vector Graphics (SVG) representation as proposed by the WorldWideWeb Consortium (W3C) may be utilized. SVG allows for three types of graphic objects: vector graphic shapes
15 (e.g., paths consisting of straight lines and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composited into previously rendered objects.

A display background (that is raster based) for the geographic data and display layers is managed as a multi-level library of raster tiles. The raster layers are
20 composed from multiple vector layers on servlet 108, resulting in better download and display performance. The raster map on the thin client 102 allows panning (virtual roaming paradigm), and zooming across multiple levels. A smart-cache on thin client 102 allows the swapping of compact tiles from the database to memory,

in a manner appropriate to the device. A single workspace per map provides the definition of the map and the display attributes for the layers on the thin client 102.

To view and interact with geographic data on a thin client 102 (such as a PDA), maps from one or more locations and associated attributes are combined
5 into a mapset. The mapset contains the geographic data in a generalized, differentially encoded, indexed vector data. The mapset is transferred from server 106 to thin client 102.

Thin client 102 users (e.g., technicians) can make scribbles and annotations on a map displayed on a thin client using a paper and pencil metaphor. To
10 accommodate such redlining, embodiments provide a redline object that consists of geometric scribbles (points, lines, polygons, symbols), GPS (global positioning system) input coordinates, annotations (positioned text with font information, etc.), and a geo-reference system that allows the redline object to be integrated into a spatial database 116 using server 106 and displayed on other maps. The redlining
15 system captures user input and allows for redline objects to be uploaded to server 106. Further, the level of sharing on the server 106 side is configurable (e.g., personal, group, global, etc.).

Markup data is uploaded from thin client 102 to server 106 for processing the markup data on the server. The processing includes creating an executable that
20 reads a database file containing the markup data (in the form of pixel coordinates) and converts or transforms the pixel coordinates to Mapping Coordinate System (MCS) and Latitude/Longitude (Lat/Lon) coordinates and outputs a standard data format (SDF) file. In the SDF file, point objects represent markup text and

polyline objects represent markup redlines. Additionally, the markup data can be geo-referenced on server 106 in a MWF (Map Window File) 124 where the original mapset (that the markup is based on) came from.

FIG. 2 illustrates a high level architecture that enables redlining and markup processing capabilities in accordance with one or more embodiments of the invention. The markup processing design consists of several client 102 and server 106 side components. The main component on the client side 102 is the synchronization application 204 that uploads the markup database (PDB) file from the thin client 102 (e.g., a personal digital assistant (PDA) device). The main component on the server side 106 is the markup publisher executable 206 that utilizes several other components such as some Java classes with an ActiveX wrapper 208, the SDF COM toolkit 210, and MapGuide ActiveX control 212. Each of these components 206-212 may reside within web server 110 and/or as part of components 108, 112-114, and 120-128.

To process thin client 102 markup data, the synchronization application 204 uploads the markup database PDB file from the thin client 102 to the server 106. The file may be uploaded to a "markups" subdirectory within the directory on the server 106 that the mapset was downloaded from. A "markups" subdirectory exists for each user of the system (e.g., within each user id). To upload the PDB file, an HTTP (HyperText Transfer Protocol) "PUT" command may be utilized. A "PUT" command provides for storing an enclosed entity under a specified URI (Uniform Resource Identifier that identifies the resource upon which to apply a request).

Markup Publisher 206 is started on the server 106 and the server 106 administrator loads the markup database PDB file that was just uploaded. From the markup publisher 206, the server 106 administrator runs the markup publisher executable 206 to begin processing the data. The executable 206 makes a call into

5 Java classes with ActiveX Wrapper 208 to convert the pixel data to MCS coordinates. Markup publisher 206 then launches an instance of a MapGuide application (e.g., using MapGuide server 120) with the same map URL (Uniform Resource Locator) where the original mapset data came from. Markup publisher 206 utilizes the instance of the MapGuide application to convert the MCS

10 coordinates to Lat/Lon coordinates. Once the data has been converted to Lat/Lon coordinates, markup publisher 206 uses the SDF COM Toolkit 210 to create an SDF file. The SDF file can be used to superimpose the geo-referenced markup entities on the original MWF 124 map. Appropriate administrator personnel on a thick client 104 or another thin client 102 can then review the markup changes.

15 FIG. 3 is a flow chart illustrating the basic data flow for processing thin client 102 markup data as described above. At step 300, the markup database PDB file is uploaded from the thin client 102 to the server 106 using the synchronization application 204. At step 302, the markup publisher 206 is started on the server 106 and at step 304, server 106 loads the markup database PDB file that was just

20 uploaded from thin client 102.

A user (or the markup publisher 206) then initiates a command (by running an executable) to begin processing the data. The command loads the Java classes with ActiveX wrapper 208. At step 306, a call is made from markup publisher 206

to a Java class to read the header and markup data from the PDB file and convert the pixel data to MCS coordinates.

Markup publisher 206 then launches an instance of a MapGuide application (with the same map URL where the original mapset data came from) at step 308.

5 Subsequently, markup publisher 206 may wait for an “onMapLoaded” message to be sent from the MapGuide application. The “onMapLoaded” message indicates when the MCS coordinates have been converted and completed loading. In response to the “onMapLoaded” message, markup publisher 206 reads the MCS coordinates from the Java classes and makes a call to the MapGuide application to
10 convert the MCS coordinates to Lat/Lon coordinates at step 310. The toolkit 210 is then utilized to create and write out the Lat/Lon coordinates to an SDF file at step 312. The SDF file can then be used to superimpose the geo-referenced markup entities on the original MWF map.

As described above, the synchronization application 204 is responsible for
15 uploading markup data from the thin client 102 to server 106. Synchronization application 204 is also responsible for deleting the markup data associated with mapsets that are no longer available on the thin client 102. The markup upload process is intertwined with the mapset download process.

FIG. 4 is a flow chart illustrating the synchronization application 204
20 markup upload process. At step 400, a thin client 102 user or application launches the synchronization application 204. At step 402, the synchronization application 204 initiates a socket connection if one is not already open. Synchronization application 204 takes inventory of the mapsets currently resident on thin client 102

at step 404. Synchronization application 204 searches for the markup data associated with the resident mapsets at step 406.

At step 408, synchronization application 204 sends all resident markup data to a specified server 106 directory (e.g., "<server-path>/<username>/markups") using HTTP "PUT" requests. If any of the PUT requests fail, the synchronization operation is terminated and an error message is displayed on the PDA 202. Synchronization application 204 downloads new mapsets at step 410 and deletes unreferenced mapsets at step 412. At step 414, synchronization application 204 deletes any markup data associated with deleted mapsets. Thereafter, synchronization application 204 returns to the calling application or terminates.

Java classes with ActiveX wrapper 208 provide the ability to read header and markup data from the PDB file and the actual markup entities and converting them to MCS coordinates. Additionally, the classes may provide public methods for accessing the data maintained within the classes through a Windows executable via the ActiveX wrapper.

Redline/Markup Features

As described above, thin clients 102 may markup the map data that is downloaded with each mapset. The markup graphics elements include a georeferenced polyline redline and a georeferenced text note. The georeferenced polyline redline is a zero width vector that mimics ink flowing from a stylus, and an associated text note that pops up as a tooltip when the object is selected. A tool

tip is a small popup window that presents a short description of a toolbar button's purpose when the mouse is positioned over a button for a period of time. The georeferenced text note provides text blocks, represented on the map as note symbols anchored at a point, and an associated text note that pops up as a tooltip when the object is selected. Any number of these objects can be included in the markup layer of a specific mapset. Each mapset references a single markup layer.

Markup layers may be provided in two forms: "correction" and "annotation". The "correction" form shows errors in the underlying map data that require action by the map authors (e.g., if a water main or other device is not located where provided in the map). The "annotation" form is used to record information that augments the map data (e.g., to indicate that a leak or power failure at a specific map location has been fixed). Both forms are stored on the server and are accessible by other map users, but only the "correction" markups are of interest to the map maintainers.

Normally, a user is redlining in "annotation" mode. When a map error is discovered, the user begins a "report map error" process/function. Thereafter, the markup screen is cleared, leaving only the base map. The user then draws markup entities to describe the problem. When finished, the user ends the "report map error" operation, possibly filling out a note/form on the overall map error. The system then returns to the "annotation" mode with all of the previous annotation entries visible.

Redline Object

FIG. 6A illustrates a redline object on a thin client 102. FIG 6B illustrates a text entry box for a redline object. The redline object 602 of FIG. 6A may be created by a user. FIG. 5 is a flow chart illustrating the creation of a redline (e.g., redline object 602) in accordance with one or more embodiments of the invention.

The redline process begins when the user is viewing a map (e.g., map 600) on a thin client 102 (e.g., PDA device). At step 500, the user selects a "new redline" map menu choice. Such a menu choice may be activated by selecting a hardware button on a PDA device, by activating a menu using a stylus, or by other common means utilized on a PDA device or thin client 102. At step 502, a redline menu is posted/used. At step 504, the user sketches on the map window 600 of thin client 102. For example, the user can drag the stylus across the screen. A line may continue to track the stylus (within a predetermined tolerance level) so long as the stylus remains in contact with the screen. To complete the redline line, the user can lift the stylus. Thereafter, the user can continue to add to the redline by pressing the stylus to the map window 600 again.

At step 506, the user chooses "finish" from the redline menu to end the redline. Upon selecting "finish", a redline text edit box 604 is displayed at step 508.

At the bottom of the redline text edit box are two buttons: "Done" 606 and "Delete" 608. The user enters any desired text at step 510. Once the user has finished entering text, the "Done" button 606 may be selected to complete the creation operation or the "Delete" button 608 can be selected to delete the redline entry at step 512. If the user presses the "Delete" button 608, a confirmation dialog

box obtains confirmation/rejection regarding whether the user wants to delete the redline entry. If no text is entered, the default text entry may comprise the text “Note” or the user’s name and the date of the entry.

In the creation process, a single “Undo” action may be supported.

- 5 However, a “redo” command may not be supported. Further, the user may not have the ability to pan or zoom while drawing a redline entry and clipboard text cut/copy/paste functions may not be supported.

To modify an existing redline, the user selects the redline object 602.

- Thereafter, a tooltip appears with as much of the text as can be shown. If
10 underlined, the text is editable. To access and edit the text, the tooltip is touched again to access text entry screen 604. The user can then edit the text and select “Done” 606 upon completion.

- To delete a redline, the user selects the desired redline object 602 on a displayed map 600. The tooltip associated with the selected redline appears
15 underlined, indicating that the tooltip is linked to something. Tapping on the tooltip text displays the redline text edit dialog box 604. The user can tap the delete button 608 in the text entry box 604 to delete the redline.

Markup Note Object

- 20 In addition to the redline line as described above, a markup note object may be placed on a map. A markup note object is a symbol that is drawn at the point where the note object is inserted. The size of the symbol is independent of the map scale.



FIG. 7 is a flow chart illustrating the creation of a markup note object in accordance with one or more embodiments of the invention. At step 700, the user selects "New Note" from the map menu in the map view screen on thin client 102. Such a menu choice may be activated by selecting a PDA device hardware button, by activating a menu using a stylus, or by other common means utilized on a PDA device or thin client 102. At step 702, the user taps on the screen at a point where the note will be anchored. A text entry screen similar to screen 604 is displayed with "Done" 606 and "Delete" 608 buttons at step 704. At step 706, the user enters the desired text. Memory restrictions may limit the amount of text to 255 characters. Once all of the desired text is entered, the "Done" button 606 is selected. Thereafter, the map view screen 600 is displayed showing an icon at the note anchor point at step 708. Similar to the redline object, a single "undo" action may be supported.

To modify a markup note, the user selects the note icon. Once selected, a tooltip appears with as much of the text as can be shown. If underlined, the text is editable. The user touches the tooltip again to access the text entry screen 604 with the text available for editing. The user can edit the text and selects the "Done" button 606 to finish and update the note text.

To delete a markup note, the user selects the desired markup note icon from map view screen 600. Once selected, a tooltip containing as much of the note text (underlined) as can be displayed appears. The note text is tapped again to gain access to text entry dialog box 604. To delete the markup note, the user merely selects the "Delete" button 608. In response, a confirmation dialog box obtains

09628851.073100

confirmation/rejection from the user to confirm deletion of the markup note.

Georeferenced Point Symbols

Georeferenced point symbols (predefined custom shapes) are a third type of
5 markup graphic element. Such point symbols are a shorthand annotation system
customized to the user application. Each symbol may be optionally associated with
a text note. For example, a georeferenced cloud line may be drawn with an
associated note text. Such a cloud may be used to call focus to a particular area on
the map.

Conclusion

This concludes the description of one or more embodiments of the invention. The following describes some alternative embodiments for accomplishing the present invention.

15 For example, any type of personal digital assistant, WINCE device, etc. can be utilized as a thin client 102. Further, any type of computer, such as a mainframe, minicomputer, or personal computer, or computer configuration, such as a timesharing mainframe, local area network, standalone personal computer, etc. could be used as a server 106.

20 In summary, a geographic information system on a personal digital assistant
or thin client provides the ability to markup or redline a map.

The foregoing description of one or more embodiments of the invention has been presented for the purposes of illustration and description. It is not

intended to be exhaustive or to limit the invention to the precise form disclosed.

Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

09628851.073100